# OmniPhotos: Casual 360° VR Photography

TOBIAS BERTEL, MINGZE YUAN, REUBEN LINDROOS, and CHRISTIAN RICHARDT, University of Bath

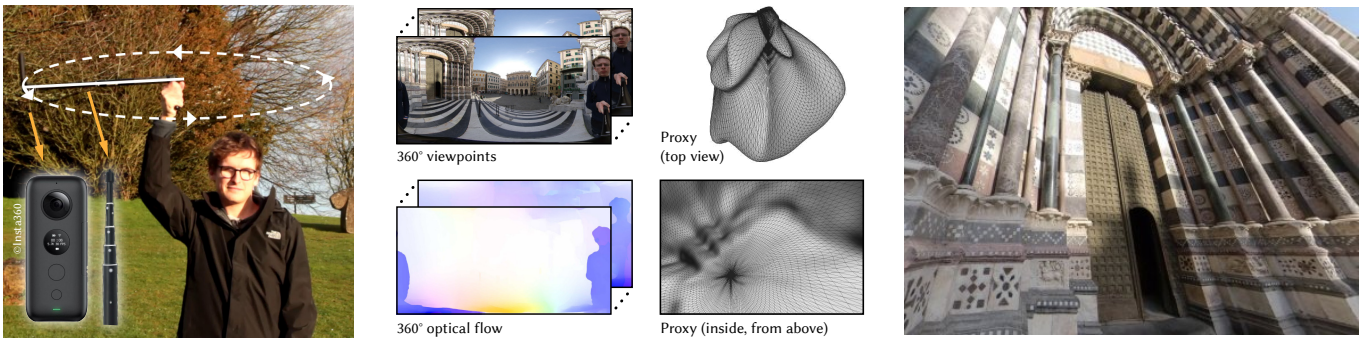



Fig. 1. OmniPhotos are 360° VR photographs that are casually captured with a single 360° video sweep. Capturing only takes 3–10 seconds and, once processed into an image-based scene representation with optical flow and scene-adaptive proxy geometry, OmniPhotos can be viewed freely in VR headsets. *Please note that this figure and others in this paper are animated; should they not be playing automatically, please consider viewing this paper with Adobe Reader.*

Virtual reality headsets are becoming increasingly popular, yet it remains difficult for casual users to capture immersive 360° VR panoramas. State-of-the-art approaches require capture times of usually far more than a minute and are often limited in their supported range of head motion. We introduce OmniPhotos, a novel approach for quickly and casually capturing high-quality 360° panoramas with motion parallax. Our approach requires a single sweep with a consumer 360° video camera as input, which takes less than 3 seconds to capture with a rotating selfie stick or 10 seconds handheld. This is the fastest capture time for any VR photography approach supporting motion parallax by an order of magnitude. We improve the visual rendering quality of our OmniPhotos by alleviating vertical distortion using a novel deformable proxy geometry, which we fit to a sparse 3D reconstruction of captured scenes. In addition, the 360° input views significantly expand the available viewing area, and thus the range of motion, compared to previous approaches. We have captured more than 50 OmniPhotos and show video results for a large variety of scenes. We will make our code available.

CCS Concepts: • **Computing methodologies** → **Computational photography**; **Image-based rendering**; **Virtual reality**.

Additional Key Words and Phrases: casual capture, image-based rendering, motion parallax, novel-view synthesis

**ACM Reference Format:**
Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. 2020. OmniPhotos: Casual 360° VR Photography. *ACM Trans. Graph.* 39, 6, Article 266 (December 2020), 12 pages. https://doi.org/10.1145/3414685.3417770

Authors' address: Tobias Bertel, T.B.Bertel@bath.ac.uk; Mingze Yuan; Reuben Lindroos; Christian Richardt, christian@richardt.name, University of Bath.


0730-0301/2020/12-ART266 $15.00
https://doi.org/10.1145/3414685.3417770

## 1 INTRODUCTION

The latest virtual reality (VR) head-mounted displays (HMDs) enable breathtaking immersion thanks to recent technological advances in near-eye display and tracking technologies [Koulieris et al. 2019]. However, capturing VR photographs that exploit the *full* immersive potential of VR, in particular including depth cues like motion parallax, is currently beyond most casual users [Richardt et al. 2019].

State-of-the-art 360° VR photography relies on panoramic light fields [Overbeck et al. 2018], which require the time-consuming capture and processing of more than a thousand input photos. This is clearly beyond the reach of casual end users. Hedman and Kopf's Instant 3D Photography approach [2018] reconstructs high-quality textured meshes from dozens of captured colour+depth images, with full 360° VR photographs requiring more than a minute of capture time. In addition, 3D reconstruction remains fragile and prone to artefacts, e.g. for thin or distant objects in a scene, such as trees. The MegaParallax approach [Bertel et al. 2019] overcomes this limitation using image-based rendering with view-dependent flow-based blending. However, the supported viewing range of motion (*aka* head box) is limited by the field of view of the used camera, and visual distortions are introduced by the basic proxy geometry. No current 360° VR photography approach simultaneously supports: (1) quick and easy capture in under 10 seconds, and (2) real-time VR rendering of 360° environments with (3) high-quality motion parallax and (4) a head box with 1 m diameter.

We introduce OmniPhotos to fill this gap – a new approach for casual 360° VR photography using a consumer 360° video camera. By attaching the 360° camera to a rotating selfie stick, as shown in Figure 1, we can significantly reduce the core capture time to less than 3 seconds, which enables rapid, casual and robust 360° VR photography. Static scenes work best, although the fast capture time reduces artefacts caused by movement in the scene. The omnidirectional view of 360° cameras also unlocks a significantly enlarged head box compared to other methods, which is ideal for

seated VR experiences. We further improve the visual fidelity of the VR viewing experience by automatically and robustly reconstructing a scene-adaptive proxy geometry that reduces vertical distortions during image-based view synthesis. We demonstrate the robustness and quality of our OmniPhotos approach on dozens of 360° VR photographs captured in seven countries across Europe and Asia. We further perform extensive ablation studies as well as quantitative and qualitative comparisons to the state of the art.

## 2 RELATED WORK

*Panoramas.* The most common type of VR photography today is 360° panoramas stitched from multiple input views [Szeliski 2006]. However, panoramas generally appear flat due their lack of depth cues like binocular disparity. This limitation is addressed by omnidirectional stereo techniques [Peleg et al. 2001; Richardt 2020], which create *stereo panoramas* from a camera moving on a circular path [Baker et al. 2020; Richardt et al. 2013], a rotating camera rig for live video streaming [Konrad et al. 2017], or per-frame from two 360° cameras [Matzen et al. 2017]. The extension of these techniques to videos using multi-camera rigs [Anderson et al. 2016; Schroers et al. 2018] is currently the standard format for 360° stereo videos. While these approaches provide stereo views with binocular disparity, most do not support motion parallax directly – the change in view as the viewpoint is moved, which is an important depth cue for human visual perception [Howard and Rogers 2008] and crucial for feeling immersed in VR [Slater et al. 1994]. Schroers et al. [2018] first demonstrated parallax interpolation for professionally captured omnistereoscopic video with a 16-camera rig.

*Panoramas with motion parallax.* Panoramas can be augmented by interactively sculpting geometry for projecting the panorama on [Sayyad et al. 2017]. Similarly, stereo panoramas can be augmented by estimating depth [Bertel et al. 2020; Thatte et al. 2016] and segmenting the panorama into multiple depth layers [Serrano et al. 2019; Zhang et al. 2020; Zheng et al. 2007], which enables free-viewpoint rendering of novel views with motion parallax. The input images can also be used directly for image-based rendering of novel views [Bertel et al. 2019; Chaurasia et al. 2013; Hedman et al. 2016; Lipski et al. 2014]. These approaches are limited to head motion in the plane of the circular camera trajectory, but using a robot arm [Luo et al. 2018], a camera gantry [Overbeck et al. 2018], or a spherical 16-camera rig [Parra Pozo et al. 2019], one can capture viewing directions over the surface of a sphere, which enables 6-degree-of-freedom (6-DoF) view synthesis using panoramic light fields. These state-of-the-art capture methods are, however, restricted to professional usage and not accessible or affordable for casual consumers interested in practising 360° VR photography. Huang et al. [2017] present an approach for mesh-based warping of 360° video according to sparse scene geometry, but the visual fidelity is limited due to warping artefacts.

*3D reconstruction.* Capturing the shape and appearance of objects or scenes by means of *3D photography* has been an active topic of research for more than 20 years [Curless et al. 2000]; we refer to Richardt et al. [2020] for an extensive review of the state of the art. Recent advances exploit the ubiquity of phone cameras for casual 3D photography [Hedman et al. 2017], and use depth maps obtained from built-in stereo cameras [Hedman and Kopf 2018; Kopf et al. 2019], multi-view stereo [Holynski and Kopf 2018], temporal stereo [Valentin et al. 2018], or monocular depth estimation [Shih et al. 2020] to reconstruct the scene geometry; similar approaches are also used to estimate depth maps from 360° images [da Silveira and Jung 2019; Im et al. 2016; Wang et al. 2020; Zioulis et al. 2019]. Most approaches produce a textured mesh as output, which can be rendered efficiently even on mobile devices, and supports motion parallax natively. For 360° VR photography, Hedman et al. [2017] use fisheye input images, which are stitched into a multilayer, textured panoramic mesh that can easily be rendered from novel views. Hedman and Kopf [2018] produce a similar output from narrow field-of-view RGBD images that are captured with minimal displacement to facilitate their registration into an RGBD panorama. Their 360° panoramic captures take around 100–200 seconds, ten times slower than our approach. Parra Pozo et al. [2019] estimate per-view depth maps using a variant of coarse-to-fine PatchMatch with temporal bilateral and median filtering. All views are rendered as a separate textured meshes and fused together using a weighting scheme. This pipeline is optimised for 6-DoF video and real-time playback. However, accurate 3D reconstruction of unconstrained environments remains challenging, particularly in uniformly coloured regions like the sky, or for highly detailed geometry such as trees. We employ image-based rendering to address these limitations and optimise for the visual fidelity of results without relying on accurate 3D reconstructions, which are hard to obtain for general scenes.

*Learned view synthesis.* Deep learning is starting to replace parts of the view synthesis pipeline or even the entire pipeline. Hedman et al. [2018] learn blending weights for view-dependent texture mapping to reduce artefacts in poorly reconstructed regions. Recently, multiplane images [Zhou et al. 2018] have set a new bar in terms of the visual quality of synthesised views from just one to four input views [Flynn et al. 2019; Mildenhall et al. 2019; Srinivasan et al. 2019; Tucker and Snavely 2020]. Concurrent work generalises this approach to multi-sphere images for rendering novel views from a 360° stereo video [Attal et al. 2020] or 46 input videos [Broxton et al. 2020], respectively. Other approaches use point clouds [Meshry et al. 2019] with deep features [Aliev et al. 2020; Wiles et al. 2020], voxel grids [Nguyen-Phuoc et al. 2019; Sitzmann et al. 2019a] or implicit functions [Mildenhall et al. 2020; Sitzmann et al. 2019b] to learn view synthesis; we refer to Tewari et al. [2020] for a recent survey on neural rendering. The main limitation of these approaches is that they do not meet the performance requirements of current VR headsets (2 views × 2 megapixels × 80 Hz = 320 MP/s), with some techniques being four orders of magnitude too slow (e.g. NeRF: $1008{\times}756/30\,s = 0.025$ MP/s). Using shaders for view-dependent texture mapping with flow-based blending, our approach consistently exceeds the required performance on an off-the-shelf laptop for a seamless, high-quality VR experience.

## 3 OMNIPHOTO PIPELINE

Our goal is to enable casual 360° VR photography of mostly static environments that is fast (less than 10 seconds), easy and robust. Our approach follows the general structure of the VR capture pipeline

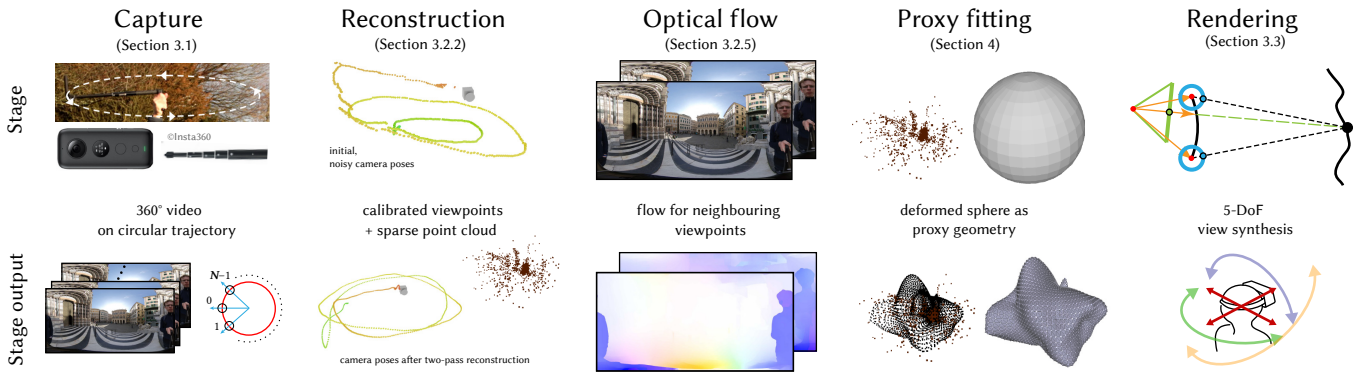



Fig. 2. Overview of the main algorithm stages and their outputs, from capture, over reconstruction, optical flow and proxy fitting, to rendering.

[Richardt et al. 2020] in terms of capture (Section 3.1), preprocessing (Section 3.2) and real-time rendering (Section 3.3). We specifically tailor the pipeline stages to optimise for casual 360° VR photography:

(1) We propose the fastest capturing procedure so far (Section 3.1) by using a consumer 360° video camera on a rotating selfie stick (although handheld capture is also possible).
(2) We introduce a scene-adaptive deformable proxy geometry fitting step in Section 4, which visibly reduces *vertical distortion* [Anderson et al. 2016; Shum and He 1999] in our results.

## 3.1 Casual capture of 360° VR photographs

The input to our approach is a single 360° video that is captured by a consumer 360° camera moving on a roughly circular path (see Figure 2). Specifically, we ensure that one of the fisheye lenses of the 360° camera is pointing radially outward, as this avoids potential stitching artefacts within the outward view. While the camera path is similar to earlier work [Bertel et al. 2019; Peleg et al. 2001; Richardt et al. 2013], there are two unique advantages to using a 360° camera instead of a normal perspective camera:

(1) The increased field of view significantly expands the supported viewing area (*aka* 'head box') for view synthesis compared to perspective input views.
(2) Thanks to the omnidirectional 360° views, most of the scene is visible in all video frames, which enables more robust camera pose estimation and scene reconstruction [Hedman et al. 2017], as inside-out perspective camera views are challenging to reconstruct with existing structure-from-motion tools [Bertel et al. 2019].

The 360° camera can be handheld, on a stretched arm, with the person rotating on the spot to capture the full 360° environment with motion parallax from multiple perspectives. This process usually takes about 10 seconds for a full rotation. We found that we can further speed up this capture process using a rotating selfie stick, to about 1.7 seconds per revolution on average. In addition, the rotating selfie stick ensures a smoother, more repeatable camera motion that is closer to an ideal circle, which reduces view interpolation artefacts in the final results. Our input video swings have an average length of 14.1±5.6 seconds, which includes set-up time, rotation speed-up, 3–5 revolutions, slow down and stopping the recording. Both capture approaches are suitable for casual users with little experience, as they are easily learned and quickly performed.

We use an 'Insta360 ONE X'[1] 360° camera for most of our results. We captured most videos at 4K (3820×1920) resolution at 50 Hz, and some videos at 3K (3008×1504) at 100 Hz or 5.7K (5760×2440) at 30 Hz to compare the trade-off between spatial resolution and the number of images per camera circle. The 4K 360° video has a resolution of 10.6 ppd (pixel per degree), which approximately matches the angular resolution of current-generation VR head-mounted displays at 11–14 ppd (e.g. Oculus Rift S, VIVE Pro); 5.7K 360° video at 16 ppd slightly exceeds current VR HMDs. We generally use an exposure time of 1/2000 seconds, or less, to minimise motion blur[2] and rolling shutter artefacts. We use automatic white-balance and an ISO level of ⩽400 to reduce noise. We observed no colour shifts due to automatic white-balancing.

## 3.2 Preprocessing of 360° VR photographs

The 360° video captured by the user in the previous section now needs to be preprocessed to enable the real-time VR rendering described in Section 3.3. This process starts with 360° video stitching and stabilisation, followed by camera reconstruction, loop selection, frame sampling, optical flow computation, and finally reconstructing our novel scene-adaptive proxy geometry.

*3.2.1 360° video stitching.* Most consumer 360° cameras record videos on-device in a proprietary format that combines the fisheye videos, audio track(s) and some metadata, such as data from built-in IMUs (inertial measurement units). These proprietary videos can then be stitched using vendor-specific software to produce 360° videos with equirectangular projection [Lee et al. 2016; Perazzi et al. 2015; Szeliski 2006], the most common monoscopic 360° video format. Working directly with stitched 360° videos means that our approach in principle supports videos stitched in any way, by any software, making it independent from any specific vendor and thus more accessible to casual users. The stitching software we use also offers a stabilised stitching option[3] that removes almost all rotational

[1] https://www.insta360.com/product/insta360-onex (last accessed 6 May 2020)
[2] Horizontal motion blur can be approximated using $\frac{\text{image-width}\times\text{exposure-time}}{\text{rotation-time}}$, which is about one pixel for a 4K video with 1/2000 s exposure time and 2 s rotation time. Slower rotations, e.g. handheld, allow for increased exposure times at the same level of blur.
[3] *Insta360 Studio 2019* calls this mode FlowState™ stabilisation. We use version 3.4.2.

camera motion while keeping vertical lines upright, presumably using IMU data recorded by the camera. This stabilisation significantly reduces the average motion magnitude between video frames, which is beneficial for tracking and optical flow estimation, as argued by Schroers et al. [2018].

*3.2.2 Camera reconstruction.* We estimate camera poses for each frame of the stitched 360° video, and reconstruct a sparse 3D point cloud of the scene using OpenVSLAM [Sumikura et al. 2019], an open-source visual SLAM approach that natively supports equirectangular 360° video. Features are tracked in an omnidirectional fashion, which helps overcome reconstruction challenges related to small-baseline normal field-of-view inside-out video inputs [Bertel et al. 2019; Hedman et al. 2017]. We perform the camera reconstruction in two passes: we first track the complete video to obtain a globally consistent 3D point cloud, and then localise all video frames with respect to the global 3D point cloud in a second pass, to obtain a globally consistent reconstruction of camera poses (see Figure 2).

*3.2.3 Loop selection.* We manually select a looping sub-clip of the video that jointly optimises the following criteria: (1) smooth camera motion over time to avoid artefacts caused by jerky motion; (2) as-continuous-as-possible looping, i.e. smooth camera motion across the cut, to prevent a visible seam in the result; and (3) if a seam is unavoidable, it should be as hidden as possible to minimise its impact, e.g. in a less interesting direction of the scene (far away or uniform textures), not 'cutting' through people. The first two criteria could be optimised automatically, but we found that the last criterion still requires manual input, so we perform the loop selection manually. Finally, we scale the global coordinate system such that the radius of the camera circle matches the measured or estimated real-world dimensions, and centre the circle at the origin without loss of generality.

*3.2.4 Frame sampling.* We observed that videos captured at 50 Hz with the rotating selfie stick produce loops of 84±14 frames (averaged over 38 videos). However, our handheld videos produce loops of 300–500 frames, depending on frame rate, as the photographer is rotating moderately slowly (~10 s per loop). To reduce space requirements and computation time in these cases, we select a subset of around 90 frames with approximately uniform angular spacing. We evaluate the impact of further downsampling to 45, 30 or 15 frames in Table 1.

*3.2.5 Optical flow.* Our view synthesis approach in Section 3.3 relies on optical flow between pairs of neighbouring images. We precompute optical flow fields using FlowNet2 [Ilg et al. 2017] and DIS flow [Kroeger et al. 2016] directly on the stitched equirectangular images. Note that these methods were designed for perspective images. They work well on the pseudo-perspective equatorial region of equirectangular images, but degrade near the poles due to the severe distortions. To ensure consistent optical flow across the azimuth wrap-around, we repeat a vertical strip of the image just beyond the left and right edges of the equirectangular projection, and crop the computed flow fields back to the original size. In practice, we find that flow fields at half the image resolution are sufficient for high-quality view synthesis at run time using view-dependent flow-based blending [Bertel et al. 2019]. Our approach is agnostic to the specific optical flow technique that is used, and thus automatically benefits from future improvements in optical flow computation techniques.

*3.2.6 Proxy fitting.* We compute a scene-adaptive proxy geometry by fitting a deformable spherical mesh to the reconstructed 3D world points in Section 4. This approach is inspired by Lee et al.'s Rich360 video stitching method [2016], which demonstrated improved alignment and blending of input videos. Our proxy fitting technique is specifically tailored for our casually captured OmniPhotos, and robustly produces scene-adaptive proxy geometry that more accurately represents the geometry of the captured scene than the simple planar or cylindrical proxy used before [Bertel et al. 2019; Richardt et al. 2013]. This step noticeably reduces visual distortions, as shown in our results.

## 3.3 Rendering 360° VR photographs

Our 360° VR photography viewer generates new viewpoints in real time given the location and orientation of the user's headset. Our rendering approach is based on the MegaParallax image-based rendering method [Bertel et al. 2019], which we extended to equirectangular images (see Figure 3a). Each desired new view $\mathbf{I}_\mathrm{D}$ is rendered by first rasterizing the proxy geometry, yielding scene points $\mathbf{X}$, and then computing the colour of each pixel $\mathbf{x}_\mathrm{D}$ independently and in parallel. Specifically, we use the direction of each pixel's camera ray $\mathbf{r}_\mathrm{D}$ in the desired output view to find the optimal input camera pair to colour the pixel, and then project the proxy 3D point $\mathbf{X}$ into both cameras using equirectangular projection giving image projections $\mathbf{x}_\mathrm{L}$ and $\mathbf{x}_\mathrm{R}$ for the left and right view, respectively. Finally, we apply MegaParallax's view-dependent flow-based blending (see Figure 3b) using the optical flow fields, $\hat{\mathbf{F}}_\mathrm{LR}$ and $\hat{\mathbf{F}}_\mathrm{RL}$, while explicitly handling the azimuth wrap-around in the flow-based blending computations. We implement our VR photography viewer using OpenVR, which at the time of writing supported a variety of consumer headsets based on SteamVR, Oculus and Windows Mixed Reality VR, with the same code base. We render stereoscopic views using the eye transformation matrices provided by OpenVR, which encode the camera poses for the left- and right-eye cameras.

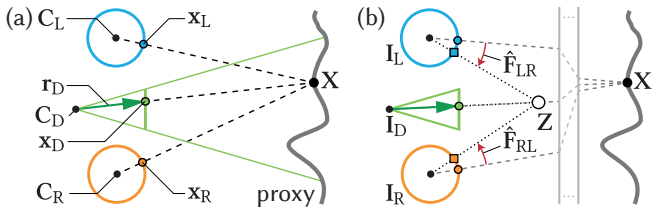



Fig. 3. Illustration of our rendering approach using equirectangular input images (shown in blue and orange). (a) Each pixel $\mathbf{x}_\mathrm{D}$ of the desired image (in green) is computed using a view-dependent blending of *two* reprojected pixel coordinates (small coloured circles) in the nearest two viewpoints. (b) We compute flow-adjusted pixel coordinates using equirectangular optical flow (small coloured squares), similar to MegaParallax [Bertel et al. 2019].

## 4 SCENE-ADAPTIVE DEFORMABLE PROXY FITTING

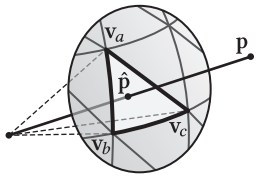
We represent the sphere mesh $\mathcal{S} = (V, F)$ in terms of vertices $V$ and triangle faces $F$. Given the spherical nature of the mesh, vertices are naturally defined in spherical coordinates $(\theta, \varphi, r)$. We initialise the vertices $V$ in a regular grid configuration of size $m \times n$, i.e. $V = \{\mathbf{v}_i\}_{i=1}^{m \times n}$, with uniform spacing along the azimuth and polar angles, and regularly tessellated triangle faces $F$. In the following, we formulate an energy minimisation that deforms this sphere mesh by adjusting the vertex radii, while keeping their angular coordinates and their triangle connectivities fixed to ensure the problem is well-conditioned and edges are not collapsing. Lee et al. [2016] found that optimising vertex radii directly may lead to unstable results with negative or very large values, which they address using additional 1D partial derivative terms. Instead, we parametrise our optimisation in terms of *inverse depth*, $d(\mathbf{p}) = 1/\|\mathbf{p}\|$, which helps regularise the scale of variables in the optimisation [Im et al. 2016], particularly for far-away points [Civera et al. 2008].

Our energy formulation consists of four terms:

$$\underset{V}{\operatorname{argmin}}\, E_{\text{data}}(P, V) + E_{\text{smooth}}(V) + E_{\text{pole}}(V) + E_{\text{prior}}(V), \quad (1)$$

where $P$ is the set of reconstructed 3D world points, and $V$ the vertices of the sphere mesh.

*Data term.* We would like to deform the sphere mesh to optimally approximate the set $P$ of 3D points, which means minimising the distance between points and triangles. By construction, as the mesh is centred at the origin, the ray from the origin through any point $\mathbf{p}$ intersects one or more triangles[4], which can be identified based on the spherical coordinates of the point $\mathbf{p}$ and the grid of vertices $V$. Let's denote the intersected triangle using $f(\mathbf{p}) = \{\mathbf{v}_a, \mathbf{v}_b, \mathbf{v}_c\}$ and the intersection point as $\hat{\mathbf{p}}$, expressed in barycentric coordinates with respect to the triangle vertices, so we can minimise the distance between all points $\mathbf{p}$ and their triangle intersections $\hat{\mathbf{p}}$:




$$E_{\text{data}}(P, V) = \frac{\lambda_{\text{data}}}{|P|} \sum_{\mathbf{p} \in P} \rho\left( \left\| d(\mathbf{p}) - d\left( \overbrace{\sum_{\mathbf{v} \in f(\mathbf{p})} b(\mathbf{p}, \mathbf{v})\mathbf{v}}^{\hat{\mathbf{p}}} \right) \right\|^2 \right), \quad (2)$$

where $b(\mathbf{p}, \mathbf{v})$ is the barycentric coordinate of $\mathbf{p}$ with respect to the vertex $\mathbf{v} \in f(\mathbf{p})$, computed in terms of the spherical angles $(\theta, \varphi)$, such that $\hat{\mathbf{p}} = \sum_{\mathbf{v} \in f(\mathbf{p})} b(\mathbf{p}, \mathbf{v})\mathbf{v}$, and $\lambda_{\text{data}}$ is the weight of the data term. In addition, we introduce a robust loss function $\rho(x)$ to make the optimisation more robust to outlier 3D points, which are unavoidable in current SLAM techniques. Specifically, we use a scaled Huber loss (with scale factor $\sigma$):

$$\rho(x) = \begin{cases} x & x \leqslant \sigma^2 \\ 2\sigma\sqrt{x} - \sigma^2 & x > \sigma^2 \end{cases} \quad (3)$$

[4]If the ray intersects an edge or a vertex, we can pick any adjacent triangle, as the resulting energy formulation is practically identical: one or two vertices will have barycentric coordinates of zero and thus not contribute to the energy.

*Smoothness term.* We use a Laplacian smoothness term to encourage smoothly varying radii within the mesh:

$$E_{\text{smooth}}(V) = \frac{\lambda_{\text{smooth}}}{|V|} \sum_{\mathbf{v} \in V} \left\| d(\mathbf{v}) - \sum_{\mathbf{w} \in N(\mathbf{v})} \frac{d(\mathbf{w})}{|N(\mathbf{v})|} \right\|^2, \quad (4)$$

where $N(\mathbf{v})$ denotes the set of vertices neighbouring $\mathbf{v}$: (1) non-polar vertices have four neighbours, along their azimuth/polar angle isocontours, and (2) polar vertices have two non-polar neighbours, on opposite sides of the sphere (same elevation, with $\Delta\text{azimuth} = \pi$). This results in 2D Laplacian losses everywhere outside the poles, and 1D Laplacian losses across both poles.

*Pole term.* In our sphere mesh representation, we have multiple vertices at the pole (the first and last 'row' of vertices correspond to the North and South pole, respectively). We constrain a pole vertex $\mathbf{v}$ and its right neighbour $\overline{\mathbf{v}}$ to be close to each other using

$$E_{\text{pole}}(V) = \frac{\lambda_{\text{smooth}}}{|V|} \sum_{\mathbf{v} \in V_{\text{poles}}} \|d(\mathbf{v}) - d(\overline{\mathbf{v}})\|^2. \quad (5)$$

*Prior term.* To handle large regions of the mesh without any 3D points, we add a weak prior term that attracts each vertex towards the mean inverse depth $d_{\text{prior}}$ of all points $P$:

$$E_{\text{prior}}(V) = \frac{\lambda_{\text{prior}}}{|V|} \sum_{\mathbf{v} \in V} \left\| d(\mathbf{v}) - d_{\text{prior}} \right\|^2. \quad (6)$$

*Implementation.* In practice, we replace each residual $\|a - b\|$ in Equations 2 and 4 to 6 with a *normalised residual*

$$\left\| \frac{a - b}{a + b} \right\| \quad (7)$$

that cancels out any global scale factor, as $\frac{(ka)-(kb)}{(ka)+(kb)} = \frac{a-b}{a+b}$. This ensures that the same globally optimal solution is found regardless of different scale factors due to varying units of length. We implement this optimisation using the Ceres non-linear least squares solver [Agarwal et al. 2012], and choose the sparse Cholesky solver to exploit the sparse structure of the energy with thousands of points. The optimisation stops when $|\Delta\text{cost}| / \text{cost} < 10^{-6}$, or after 100 iterations. For the initial solution, we set all vertices to the mean inverse depth of all points; more sophisticated schemes like a hemisphere with a ground plane are possible. We evaluate a range of parameter values in Figure 9 and Table 1, and use the following parameter values for all our results: $m = 160$, $n = 80$, $\lambda_{\text{data}} = 1$, $\sigma = 0.1$, $\lambda_{\text{smooth}} = 100$, $\lambda_{\text{prior}} = 0.001$.

## 5 RESULTS AND EVALUATION

Figure 4 shows 30 OmniPhotos we captured and processed using our approach. Three of these were taken handheld (Cathedral, Shrines 1+2), with the majority (90%) captured using our rotating selfie stick with an average loop length of 1.2–1.8 seconds. The selfie stick is telescopic, which allows for capture radii between 33 and 100 cm, with about 63% at 55 cm and 27% at 78 cm.

In this section, we show qualitative results and comparisons, perform quantitative evaluation and ablation studies, and finally discuss the computational performance of our approach. Our results are best appreciated and evaluated in motion, which gives a better

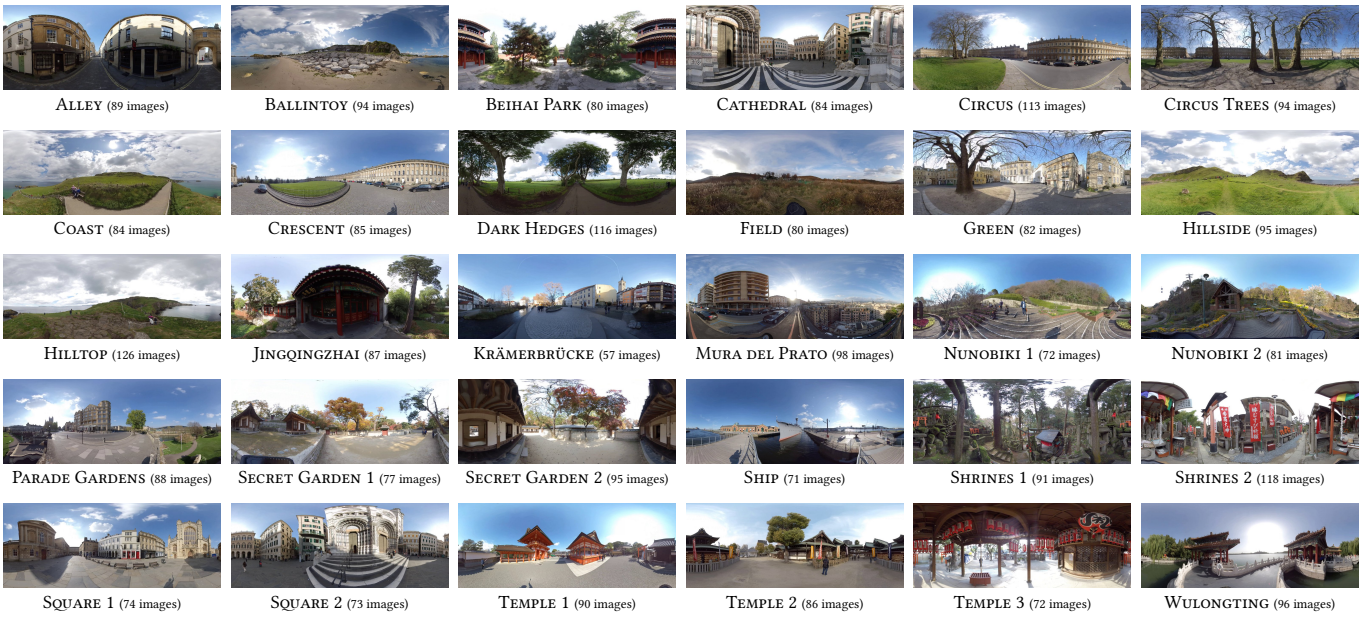



Fig. 4. Datasets shown in our paper and supplemental material. Slightly cropped for visualisation.

impression of the visual experience. To this end, we include some *animated figures* in our paper that can be viewed using Adobe Reader. We further include extensive visual results and comparisons in our supplemental material and video.

## 5.1 Comparative evaluation

The approaches closest to ours, Bertel et al.'s MegaParallax [2019] and Luo et al.'s Parallax360 [2018], also use image-based rendering with flow-based blending to synthesise novel views in real time. However, they rely on basic proxy geometry, which causes vertical distortion artefacts in nearby regions, as illustrated in Figure 5. Our scene-adaptive deformable proxy geometry deforms to fit the scene more closely, which greatly reduces these vertical distortion artefacts, as visible in Figure 6 and our supplemental material.

We next compare to Casual 3D Photography [Hedman et al. 2017]. Their 360° 3D photos were reconstructed from around 50 fisheye DSLR photos, which take about one minute to capture, an order of magnitude slower than our approach. Their 3D reconstruction approach works well for textured scenes, but fails for fine geometry like tree branches, or uniformly coloured regions like the sky, for which accurate depth estimation and 3D reconstruction remain open problems. As their implementation is not available but their datasets are, we process one of their two camera circles (about 25 images) with our approach. To adapt their fisheye images to our approach, we first undistort them to equirectangular images and then stabilise the views by rotating them inversely to the camera orientations. Figure 7 shows that our image-based rendering approach does not require a highly accurate 3D reconstruction for convincing view synthesis from the same input. Monocular 3D photography approaches [Kopf et al. 2019; Shih et al. 2020] also tend to fail for complex geometry, as shown in Figure 8. Our OmniPhotos achieve better visual results

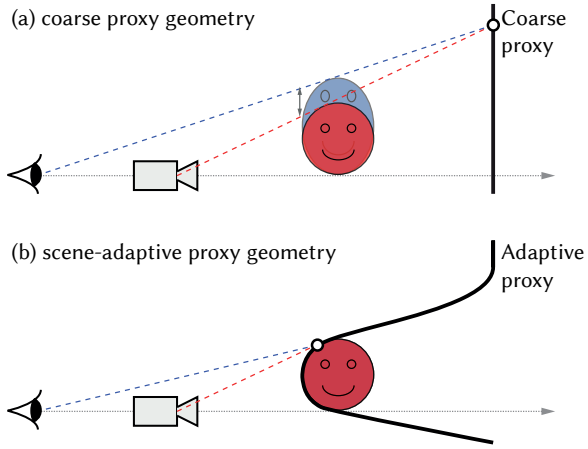



Fig. 5. Coarse proxy geometry (a) introduces vertical distortion as the input cameras are closer to the object than the viewing location (the eye). The red face, as seen by the camera, appears vertically stretched (blue face) when rendered using the coarse proxy geometry for a viewpoint behind the camera. (b) Our scene-adaptive proxy geometry deforms to fit the scene better, which strongly reduces vertical distortion.

thanks to multi-view input and the combination of scene-adaptive proxy geometry and flow-based blending for aligning texture details.

Our next comparison is to Serrano et al.'s approach for adding motion parallax to 360° videos captured with a static camera [2019]. As their approach takes as input a 360° RGBD video, we render an equirectangular image and depth map from Hedman et al.'s datasets using Blender and repeat this 360° RGBD frame to create a (static) 360° RGBD video. The resulting static scene does not play to their

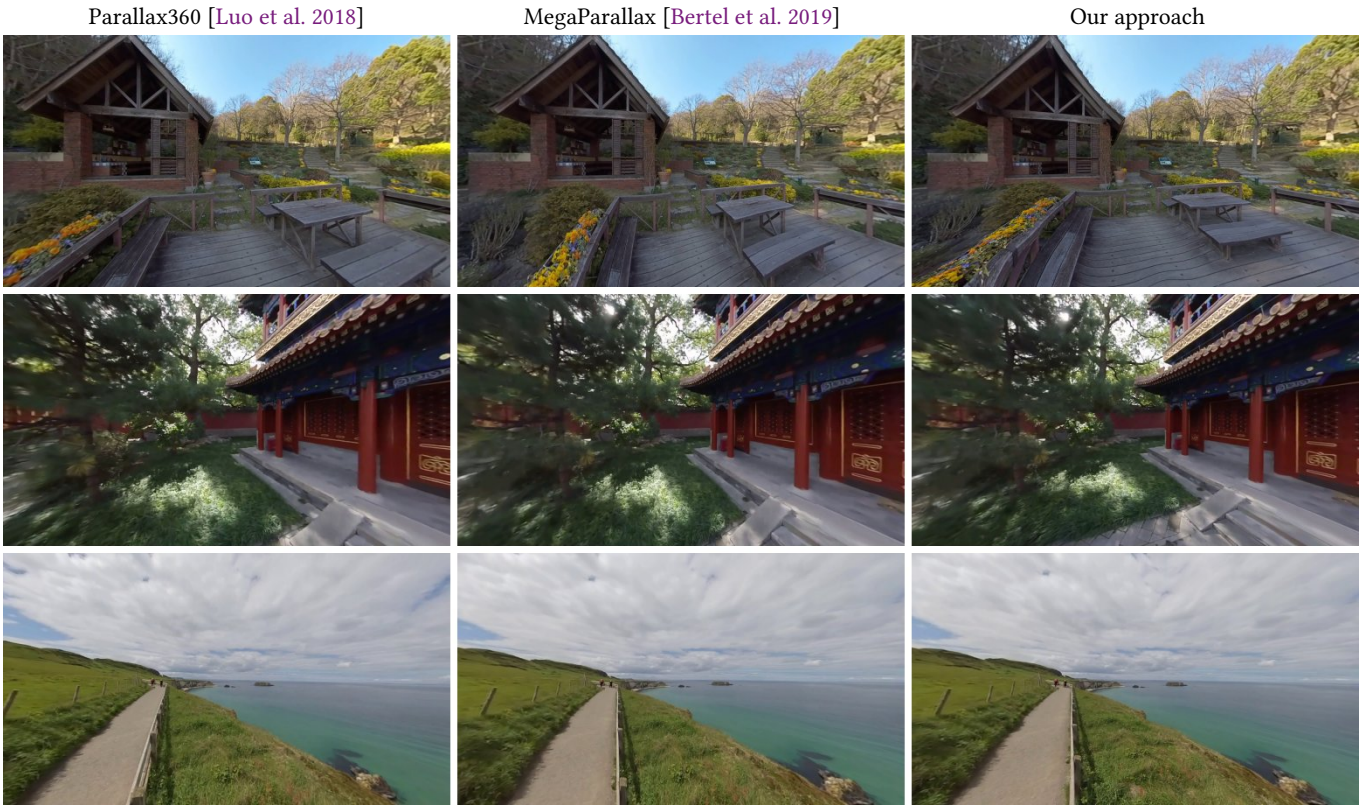



Fig. 6. Comparison of image-based 360° VR photography techniques for a virtual camera moving on a circular path. Our result reduces vertical distortion visibly, as can be seen in the table benches in the top row. **This is an animated figure**, please view with Adobe Reader if it does not play. Parallax360 [Luo et al. 2018] interpolates views on the capture circle, but not inside of it for the virtual camera path. MegaParallax [Bertel et al. 2019] generates views that suffer from vertical distortion, which distorts motion parallax. Our results show clear improvements in the quality of view synthesis and motion parallax.


Fig. 7. Comparison to Hedman et al.'s Casual 3D Photography [2017] and Serrano et al.'s Motion Parallax for 360° RGBD Video [2019] on two datasets from Hedman et al. [2017]. 3D reconstruction works well for the highly textured Library scene (top), but struggles with the thin tree branches and distant clouds in the BoatShed scene (bottom). Green regions are holes in the textured mesh. For Serrano et al.'s approach, we use colour and depth from Hedman et al.'s results, which works well for foreground objects with accurate depth, but not for occluded regions that are challenging to fill from the monocular 360° input. Our approach works well for both datasets, but shows some flow warping artefacts due to the undersampled input views (only 25 views).

Table 1. Quantitative comparison of baseline methods (top) and ablated versions of our approach (bottom). Numbers are mean±standard error; '▲' means higher is better, '▼' means lower is better. 'GT' indicates ground truth, and '*' a modified proxy geometry. Please see Section 5.2 for a detailed description.

| Baseline/Ablation Model | Images | Proxy | LPIPS▼ | SSIM▲ | PSNR▲ |
|---|---|---|---|---|---|
| MegaParallax [Bertel et al. 2019] | 90 | cylinder | 0.169±0.002 | 0.750±0.003 | 21.83±0.12 |
| MegaParallax [Bertel et al. 2019] | 90 | plane | 0.181±0.002 | 0.737±0.003 | 21.45±0.12 |
| Parallax360 [Luo et al. 2018] | 90 | cylinder | 0.207±0.003 | 0.711±0.003 | 20.75±0.11 |
| Our complete method | 90 | ours | 0.059±0.001 | 0.867±0.002 | 28.02±0.09 |
| 0) Our method (ground-truth inputs) | 90 | GT | 0.041±0.000 | 0.905±0.001 | 30.08±0.11 |
| 1) No robust data term | 90 | ours* | 0.062±0.001 | 0.859±0.002 | 27.64±0.10 |
| 2) No normalised residuals | 90 | ours* | 0.072±0.001 | 0.854±0.002 | 27.30±0.10 |
| 3) Optimising depth + no normalised residuals | 90 | ours* | 0.073±0.001 | 0.853±0.002 | 27.28±0.10 |
| 4) Optimising depth (not inverse) | 90 | ours* | 0.059±0.001 | 0.867±0.002 | 28.01±0.10 |
| 5) DIS flow [Kroeger et al. 2016] | 90 | ours | 0.060±0.001 | 0.865±0.002 | 27.98±0.09 |
| 6) No flow (linear blending) | 90 | ours | 0.059±0.001 | 0.868±0.002 | 28.03±0.09 |
| 7a) Low-resolution proxy ($m=80$, $n=40$) | 90 | ours* | 0.067±0.001 | 0.843±0.002 | 27.07±0.09 |
| 7b) High-resolution proxy ($m=240$, $n=120$) | 90 | ours* | 0.064±0.001 | 0.867±0.002 | 27.78±0.10 |
| 8a) Less smooth ($\lambda_{\text{smooth}}=10$) | 90 | ours* | 0.068±0.001 | 0.866±0.002 | 27.70±0.10 |
| 8b) More smooth ($\lambda_{\text{smooth}}=1000$) | 90 | ours* | 0.064±0.001 | 0.849±0.002 | 27.31±0.09 |
| 9a) Fewer images (1 view per 8°) | 45 | ours | 0.061±0.001 | 0.864±0.002 | 27.96±0.09 |
| 9b) Fewer images (1 view per 12°) | 30 | ours | 0.063±0.001 | 0.862±0.002 | 27.90±0.09 |
| 9c) Fewer images (1 view per 24°) | 15 | ours | 0.071±0.001 | 0.855±0.002 | 27.44±0.09 |

method's strength of propagating background information behind dynamic objects. Please see Figure 7 and our supplemental video.

## 5.2 Quantitative evaluation

We quantitatively evaluate and compare our OmniPhotos approach to the most closely-related baseline methods [Bertel et al. 2019; Luo et al. 2018], and validate our design choices and parameters using an extensive ablation study in Table 1. We perform this evaluation in the spirit of virtual rephotography [Waechter et al. 2017] on a synthetic test set of five scenes (Apartment0, Hotel0, Office0, Room0, Room1) from the Replica dataset [Straub et al. 2019]. Specifically, we render synthetic equirectangular images on a camera circle with a radius of 0.5 m as input for the various methods, and we evaluate cubemap views generated by each baseline/ablation at 69 locations inside the capture circle, on a 10 cm Cartesian grid. We do not evaluate the up/down views to focus our evaluation on the region near the equator, where viewers tend to fixate when exploring panoramas [Sitzmann, Serrano et al. 2018]. For each location, we render 512×512 cube maps, and compare the generated view to the ground truth using structural similarity index (SSIM; Wang et al., 2004), peak signal-to-noise ratio (PSNR), and the LPIPS perceptual similarity measure [Zhang et al. 2018]. We report the maximum value within a shiftable window of ±1 pixel. Note that this evaluation uses indoor spaces whereas our real OmniPhotos were all captured outdoors (Figure 4).

Our OmniPhotos quantitatively outperform MegaParallax and Parallax360 by a large margin, in addition to the clear qualitative improvement visible in Figure 6 and our supplemental material. We next evaluate our method on ground-truth camera poses and proxy geometry (0) to test the upper limit of our approach. In the next rows, we replace our robust data term with a plain L2 loss (1), remove our normalised residuals (2), and use depth instead of inverse depth (4), each of which reduces performance. Using depth instead of inverse depth (3), DIS flow (5) or no flow (6), achieves comparable performance to our approach. Row 3 shows that depth and inverse depth perform similarly when using normalised residuals. This suggests that using inverse depth and using normalised residuals are complimentary techniques for regularising the scale of variables during the optimisation. The normalised residuals have the additional benefit that one set of parameter values works for both depth


Fig. 8. Current 3D photography approaches, such as Shih et al.'s, struggle with complex scenes like the pillars (left), as well as fine geometry, like leaves (centre) or a rope (right). Our approach succeeds due to our image-based rendering approach. Please see the **animated figure** for full effect.

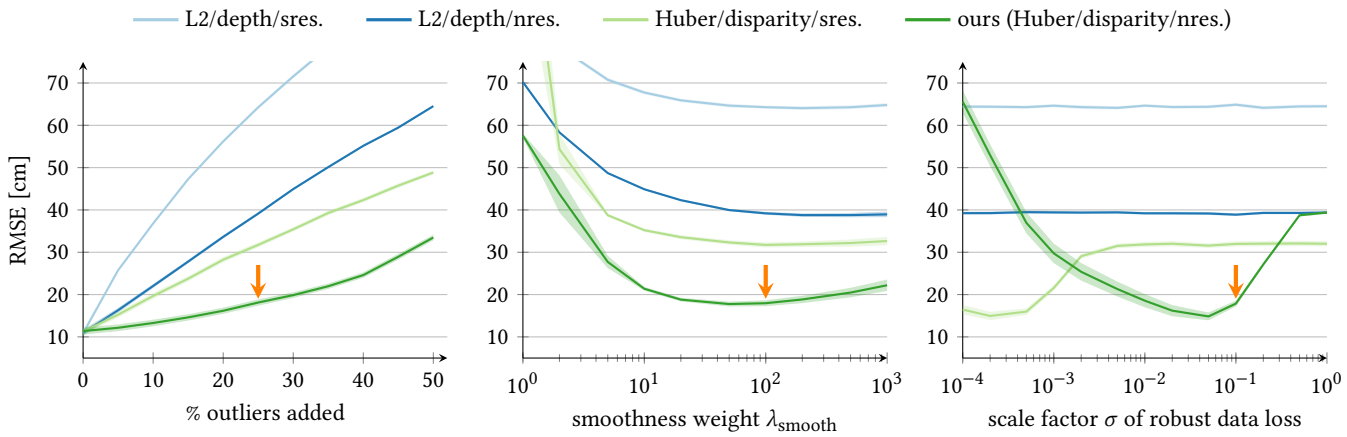



Fig. 9. Evaluation of robustness and parameter choices for different versions of our scene-adaptive deformable proxy fitting on five ground-truth scenes (Apartment0, Hotel0, Office0, Room0, Room1) from Replica [Straub et al. 2019]. We measure reconstruction accuracy using RMSE in cm, see Section 5.2.1 for details. The shaded areas indicate the standard error of the mean. We compare Huber versus L2 data loss (Equation 2), optimisation in terms of depth or inverse depth (disparity), and standard residuals ('sres.') versus our normalised residuals ('nres.', Equation 7). **Left:** Our proxy fitting technique (dark green line) is the most robust to an increasing number of outlier 3D points. The arrow indicates the level of outliers we assume for the following comparisons. **Centre and right:** Our chosen smoothness weight of $\lambda_{\text{smooth}} = 100$ and robust loss scale factor $\sigma = 0.1$ (indicated by arrows) are close to the global minimum reconstruction errors, and empirically work better for outdoor scenes that have more depth complexity than the indoor rooms of Replica. The light green line shows that standard residuals do work in practice, but the optimal value of the robust loss scale factor $\sigma$ will depend on the scale of the scene.

and inverse depth, despite their scale differences. Changing the resolution (7) or smoothness (8) of the proxy geometry results in a drop in performance. Reducing the number of input views (9) steadily reduces performance, with 45 input images almost matching the performance of 90 input views.

*5.2.1 Proxy accuracy.* In addition to the visual quality of generated views, we also evaluate the accuracy of our deformable proxy fitting in Figure 9. This experiment evaluates the robustness and parameter choices for different versions of our scene-adaptive deformable proxy fitting on five ground-truth scenes from the Replica dataset [Straub et al. 2019]. We render 1920×960 synthetic equirectangular depth maps and downsample them using area averaging to $80 \times 40 = 3200$ 3D points, to approximately match the number of 3D points we usually obtain from OpenVSLAM [Sumikura et al. 2019]. To simulate typical SLAM noise and outliers, we add ±2 cm uniform noise to all 3D point locations, and add 25%=800 outlier points sampled from a 10-metre cube centred on the scene. We measure the reconstruction quality of the proxy geometry using RMSE per vertex of the spherical depth map, in cm, averaged over 10 runs for each of the five scenes. Figure 9 shows that our proposed approach, with robust Huber data loss on inverse depth and normalised residuals, performs best with increasing number of outliers. Our default parameter values, which we use for all our OmniPhotos, can also be seen to produce results close to the global minimum, in terms of reconstruction error, within the explored design space. We also observed that the quality of the proxy geometry increases with the number of (inlier) scene points that can be used to guide the deformation process, for example using sparse COLMAP reconstructions [Schönberger and Frahm 2016] or dense multi-view stereo reconstructions [Parra Pozo et al. 2019].

## 5.3 Performance

Freshly captured OmniPhotos can be processed in about 30–40 minutes on a standard computer (3 GHz 8-core CPU, 16 GB RAM, NVIDIA GeForce RTX 2060). For a typical 9-second 360° video with 3840×1920 at 50 Hz (450 frames total, 90 frame loop), these are the major preprocessing steps:

- Stabilised 360° video stitching with CUDA: ~12 seconds
- Two-pass OpenVSLAM reconstruction: ~3 minutes
- Blender visualisation import: ~15 minutes
- Manual loop selection: ~5 minutes
- Reading images & other IO: ~20 seconds
- Scene-adaptive proxy fitting: ~10 seconds
- FlowNet2 / DIS flow: ~10 minutes / ~20 seconds

Importantly, the reconstruction with OpenVSLAM is about two orders of magnitude faster than with COLMAP. The unoptimised size of preprocessed OmniPhotos is dominated by the precomputed optical flow fields (14 MB/frame), followed by the input images (~2 MB/frame) and the proxy geometry (0.8 MB). For a typical dataset with 90 frames, this sums up to about 1.4 GB all-in. Our viewer loads such a dataset from SSD into GPU memory in about 20 seconds. Rendering of 1920×1080 views consistently takes less than 4.16 ms (240 Hz), and VR rendering is performed at the 80 Hz display rate of an Oculus Rift S HMD, for a smooth and immersive VR experience.

# 6 DISCUSSION

*Applications.* OmniPhotos are a great new way to *reliably* capture immersive 3D environments for casual to ambitious consumers as well as professional users. OmniPhotos can capture personal memories, for example on holidays, or group photos on family occasions. It would be interesting to see how people could create stories by concatenating multiple OmniPhotos. In terms of professional

applications, OmniPhotos are ideal for virtual tourism, which lets people explore far-away places from the comfort of their own home. OmniPhotos would also be useful for real estate scenarios to capture outdoor spaces or individual rooms.

*Resolution vs frame rate.* As discussed in Section 3.1, we captured input videos with different resolutions and frame rates to evaluate the trade-off between spatial resolution and the number of images per camera circle. We were originally aiming to capture more than 100 views per camera circle, but our new scene-adaptive proxy geometry has significantly reduced the number of required input views from 200–400 [Bertel et al. 2019] to 50–100 for our approach (see Table 1, row 9). Visually, the 5.7K videos produce the highest-fidelity VR photos, even when downsampled to 4K. The native 4K resolution tends to be slightly blurry, as it is the result of stitching two 2K×2K fisheye images into a 4K×2K equirectangular image. Finally, the 3K videos look noticeably blurry in the final result.

*Viewing area analysis.* Our rendering approach is modelled after MegaParallax [Bertel et al. 2019] and we can therefore benefit from their theoretical analysis of the supported viewing area (aka head box). They showed that the horizontal translation $x$ is limited to $x < r \sin \frac{\gamma}{2}$ for a given camera circle radius $r$ and camera field of view $\gamma$. The field of view of our cameras is effectively $\gamma = \pi$, as they capture the complete outward-facing hemisphere. This yields the radius of the camera circle as the upper limit of the viewing space radius. Experiments verify this behaviour, our synthesis works anywhere inside the camera circle, i.e. most of our OmniPhotos provide a head box with 1-metre diameter (capture radius: 55 cm). Schroers et al. [2018] also analysed the minimum visible depth observed by two cameras in a circular configuration. Their formula is expressed in terms of the field of view $\gamma = \pi$ and the angle $\theta$ between optical axes of adjacent cameras ($\theta \approx \frac{2\pi}{N}$ for $N$ cameras):

$$d = r \frac{\sin(\pi - \gamma/2)}{\sin(\gamma/2 - \theta)} = \frac{r}{\cos\left(\frac{2\pi}{N}\right)}. \tag{8}$$

For $N = 90$ cameras, like in our case, this evaluates to 0.24% of the capture circle radius, or 1.3 mm for $r = 55$ cm, which is negligible.

*Compression.* OmniPhotos can be compressed from 1.4 GB to a more reasonable 0.25 GB (18%) using off-the-shelf 7-Zip. A further 0.07 GB can be saved if optical flow fields are not transmitted and instead computed on the local machine (final size: 0.18 GB or 13%).

## 6.1 Limitations and future work

All approaches have limitations; we discuss the most important ones here and use them to motivate directions for future work.

*Proxy geometry.* While deforming a sphere mesh to fit *into* the reconstructed point cloud usually works well in practice (see Figure 6), it clearly has its limitations. Its fixed topology combined with the enforced smoothness produces a relatively smooth proxy geometry, which can cause warping artefacts in areas with large depth differences. Object boundaries of nearby objects, essential for (dis-)occlusion effects, cannot be fitted tightly enough, leading to warping artefacts that tend to change as the viewpoint changes (see Figure 10). These issues could potentially be overcome in different ways: (1) Mesh vertices could be moved more freely, not just radially, e.g. to align to depth edges. (2) Multi-view stereo or optical flow correspondences would provide more scene points that can make the proxy geometry more accurate and detailed. (3) Learned methods like monocular depth estimation [e.g. Wang et al. 2020] or implicit scene representations [e.g. Mildenhall et al. 2019] could be used to densify sparse reconstructions, especially in texture-less regions. As demonstrated by the ground-truth proxy experiment in Table 1, better proxy geometry improves visual results, as expected.

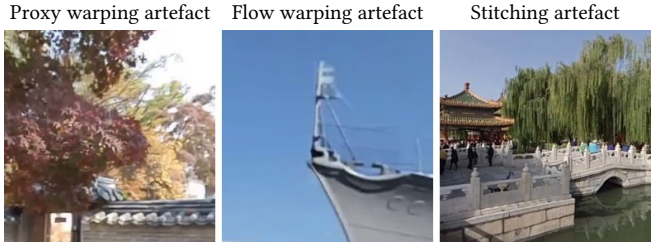



Fig. 10. Remaining visual artefacts in our results. Errors in the proxy geometry or optical flow may produce warping artefacts. We observed proxy warping artefacts primarily at large depth discontinuities, while most flow warping artefacts affect objects adjacent to a uniform region like the sky. A stitching bug in the Insta360 Studio software causes a 'swimming' artefact.

*Optical flow.* Even though the quantitative evaluation in Table 1 may suggest otherwise, flow-based blending helps reduce ghosting artefacts when the scene proxy does not fit the real scene geometry tightly. Examples for this include detailed geometry, like fences or thin tree branches (Figure 8), or reflections, for which there is a mismatch between the real and apparent depth. In some cases, we observed that FlowNet2 predicted incorrect flow near strong edges, e.g. a ship vs the blue sky (see Figure 10), which results in view interpolation artefacts. In these cases, we fall back to DIS flow.

*Stitching artefacts.* We observed minor to moderate stitching artefacts being introduced in some videos, particularly those captured at 3K/100 Hz. These artefacts are not limited to the overlap region between the two fisheye lenses and appear to be caused by warping parts of the video frame incorrectly, probably due to a software bug.[5] Since the artefacts are not consistent over time, they can cause 'swimming' during rendering, as shown in Figure 10. We only found these artefacts in the stabilised stitch, not the standard stitch. However, we consider the benefits of the stabilised stitch (improved camera reconstruction and flow computation) to outweigh these usually minor artefacts in some of our OmniPhotos.

*Vertical motion.* Our approach provides compelling 5-degree-of-freedom (5-DoF) view synthesis by supporting arbitrary head rotations as well as translations in the plane of the capture circle (see Figure 2). The missing DoF is vertical translation as our capture approach deliberately captures viewpoints at roughly the same height and thus cannot plausibly synthesise new viewpoints from a different height. In practice, this is not a problem for seated VR experiences, where users naturally keep their heads at a consistent

[5] This bug in the proprietary software Insta360 Studio 3.4.2 has been fixed in v3.4.10.

height. Capturing camera views on a sphere instead of a circle can overcome this limitation [Luo et al. 2018; Overbeck et al. 2018].

*Memory footprint.* Our uncompressed OmniPhotos require more than one GB of memory, which is manageable for a 360° VR photo experience, but cannot be easily extended to 360° VR video. By far the largest contributor to this memory footprint are the precomputed optical flow fields. Reducing the number of input views can reduce the memory footprint, and so can discarding the inward-facing hemisphere of the input images and their flow fields. In many cases, the proxy geometry aligns the input views sufficiently well without optical flow. In these regions, no flow needs to be stored, which could lead to a more compact scene-dependent flow storage format.

*Editing.* Our OmniPhotos are currently limited to reproducing the scenes that were captured as is. Virtual objects, such as digital humans, can easily be rendered on top, but the quality of occlusions by scene geometry, such as trees or buildings, is limited by the detail of the proxy geometry. Relighting the captured scene, adding new objects with consistent lighting, or removing captured objects are interesting directions for future work.

*Combination of proxy and flow.* For future work, we would like to investigate the design space of camera poses, proxy geometry and optical flow with respect to the observed visual artefacts in the rendered results. A promising direction might be a differentiable renderer for jointly optimising scene and camera geometry as well as flows to maximise the quality of synthesised views.

## 7 CONCLUSION

We presented OmniPhotos, a new type of 360° VR photography that enables fast, casual and robust capture of immersive real-world VR experiences. The key to the fast capture of OmniPhotos is to rotate a consumer 360° video camera mounted on a rotary selfie stick, which takes less than 3 seconds per loop or 10 seconds overall, and is currently the fastest approach for capturing immersive 360° VR photos. The visual quality of our novel view rendering is significantly improved by the automatic reconstruction of a scene-adaptive deformable proxy geometry, which reduces the number of required input views by a factor of 4 and strongly reduces vertical distortion compared to the state of the art. Our approach robustly creates OmniPhotos across a wide range of outdoor scenes, as demonstrated in our results and supplemental material. We will publicly release our OmniPhotos implementation in the hope of enabling casual consumers and professional users to create and experience their own OmniPhotos.

## ACKNOWLEDGMENTS

We thank the reviewers for their thorough feedback that has helped to improve our paper. We also thank Peter Hedman, Ana Serrano and Brian Cabral for helpful discussions, and Benjamin Attal for his layered mesh rendering code.

This work was supported by EU Horizon 2020 MSCA grant FIRE (665992), the EPSRC Centre for Doctoral Training in Digital Entertainment (EP/L016540/1), RCUK grant CAMERA (EP/M023281/1), an EPSRC-UKRI Innovation Fellowship (EP/S001050/1), a Rabin Ezra Scholarship and an NVIDIA Corporation GPU Grant.

## REFERENCES

Sameer Agarwal, Keir Mierle, and Others. 2012. Ceres Solver. http://ceres-solver.org.

Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. 2020. Neural Point-Based Graphics. In *ECCV*. DOI: 10.1007/978-3-030-58542-6_42

Robert Anderson, David Gallup, Jonathan T. Barron, Janne Kontkanen, Noah Snavely, Carlos Hernandez, Sameer Agarwal, and Steven M. Seitz. 2016. Jump: Virtual Reality Video. *ACM Transactions on Graphics* 35, 6 (2016), 198:1–13. DOI: 10.1145/2980179.2980257

Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. 2020. MatryODShka: Real-time 6DoF Video View Synthesis using Multi-Sphere Images. In *ECCV*. DOI: 10.1007/978-3-030-58452-8_26

Lewis Baker, Steven Mills, Stefanie Zollmann, and Jonathan Ventura. 2020. CasualStereo: Casual Capture of Stereo Panoramas with Spherical Structure-from-Motion. In *IEEE VR*. DOI: 10.1109/VR46266.2020.00102

Tobias Bertel, Neill D. F. Campbell, and Christian Richardt. 2019. MegaParallax: Casual 360° Panoramas with Motion Parallax. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (2019), 1828–1835. DOI: 10.1109/TVCG.2019.2898799

Tobias Bertel, Moritz Mühlhausen, Moritz Kappel, Paul Maximilian Bittner, Christian Richardt, and Marcus Magnor. 2020. Depth Augmented Omnidirectional Stereo for 6-DoF VR Photography. In *IEEE VR Posters*. DOI: 10.1109/VRW50115.2020.00181

Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive Light Field Video with a Layered Mesh Representation. *ACM Transactions on Graphics* 39, 4 (2020), 86:1–15. DOI: 10.1145/3386569.3392485

Gaurav Chaurasia, Sylvain Duchêne, Olga Sorkine-Hornung, and George Drettakis. 2013. Depth Synthesis and Local Warps for Plausible Image-based Navigation. *ACM Transactions on Graphics* 32, 3 (2013), 30:1–12. DOI: 10.1145/2487228.2487238

Javier Civera, Andrew J. Davison, and J. M. Martínez Montiel. 2008. Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics* 24, 5 (2008), 932–945. DOI: 10.1109/TRO.2008.2003276

Brian Curless, Steve Seitz, Jean-Yves Bouguet, Paul Debevec, Marc Levoy, and Shree K. Nayar. 2000. 3D Photography. In *SIGGRAPH Courses*. http://www.cs.cmu.edu/~seitz/course/3DPhoto.html

Thiago Lopes Trugillo da Silveira and Claudio R Jung. 2019. Dense 3D Scene Reconstruction from Multiple Spherical Images for 3-DoF+ VR Applications. In *IEEE VR*. 9–18. DOI: 10.1109/VR.2019.8798281

John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. 2019. DeepView: View Synthesis With Learned Gradient Descent. In *CVPR*. 2367–2376. DOI: 10.1109/CVPR.2019.00247

Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. 2017. Casual 3D Photography. *ACM Transactions on Graphics* 36, 6 (2017), 234:1–15. DOI: 10.1145/3130800.3130828

Peter Hedman and Johannes Kopf. 2018. Instant 3D Photography. *ACM Transactions on Graphics* 37, 4 (2018), 101:1–12. DOI: 10.1145/3197517.3201384

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep Blending for Free-Viewpoint Image-Based Rendering. *ACM Transactions on Graphics* 37, 6 (2018), 257:1–15. DOI: 10.1145/3272127.3275084

Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. 2016. Scalable Inside-Out Image-Based Rendering. *ACM Transactions on Graphics* 35, 6 (2016), 231:1–11. DOI: 10.1145/2980179.2982420

Aleksander Holynski and Johannes Kopf. 2018. Fast Depth Densification for Occlusion-aware Augmented Reality. *ACM Transactions on Graphics* 37, 6 (2018), 194:1–11. DOI: 10.1145/3272127.3275083

Ian P. Howard and Brian J. Rogers. 2008. *Seeing in Depth*. Oxford University Press. DOI: 10.1093/acprof:oso/9780195367607.001.0001

Jingwei Huang, Zhili Chen, Duygu Ceylan, and Hailin Jin. 2017. 6-DOF VR videos with a single 360-camera. In *IEEE VR*. 37–44. DOI: 10.1109/VR.2017.7892229

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *CVPR*. DOI: 10.1109/CVPR.2017.179

Sunghoon Im, Hyowon Ha, François Rameau, Hae-Gon Jeon, Gyeongmin Choe, and In So Kweon. 2016. All-around Depth from Small Motion with A Spherical Panoramic Camera. In *ECCV*. DOI: 10.1007/978-3-319-46487-9_10

Robert Konrad, Donald G. Dansereau, Aniq Masood, and Gordon Wetzstein. 2017. SpinVR: Towards Live-Streaming 3D Virtual Reality Video. *ACM Transactions on Graphics* 36, 6 (2017), 209:1–12. DOI: 10.1145/3130800.3130836

Johannes Kopf, Suhib Alsisan, Francis Ge, Yangming Chong, Kevin Matzen, Ocean Quigley, Josh Patterson, Jossie Tirado, Shu Wu, and Michael F. Cohen. 2019. Practical 3D Photography. In *CVPR Workshops*.

George Alex Koulieris, Kaan Akşit, Michael Stengel, Rafał K. Mantiuk, Katerina Mania, and Christian Richardt. 2019. Near-Eye Display and Tracking Technologies for Virtual and Augmented Reality. *Computer Graphics Forum* 38, 2 (2019), 493–519. DOI: 10.1111/cgf.13654

Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. 2016. Fast Optical Flow Using Dense Inverse Search. In *ECCV*. 471–488. DOI: 10.1007/978-3-319-46493-0_29

Jungjin Lee, Bumki Kim, Kyehyun Kim, Younghui Kim, and Junyong Noh. 2016. Rich360: Optimized Spherical Representation from Structured Panoramic Camera Arrays. *ACM Transactions on Graphics* 35, 4 (2016), 63:1–11. DOI: 10.1145/2897824.2925983

Christian Lipski, Felix Klose, and Marcus Magnor. 2014. Correspondence and Depth-Image Based Rendering a Hybrid Approach for Free-Viewpoint Video. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 6 (2014), 942–951. DOI: 10.1109/TCSVT.2014.2302379

Bicheng Luo, Feng Xu, Christian Richardt, and Jun-Hai Yong. 2018. Parallax360: Stereoscopic 360° Scene Representation for Head-Motion Parallax. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1545–1553. DOI: 10.1109/TVCG.2018.2794071

Kevin Matzen, Michael F. Cohen, Bryce Evans, Johannes Kopf, and Richard Szeliski. 2017. Low-cost 360 Stereo Photography and Video Capture. *ACM Transactions on Graphics* 36, 4 (2017), 148:1–12. DOI: 10.1145/3072959.3073645

Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. 2019. Neural Rerendering in the Wild. In *CVPR*. DOI: 10.1109/CVPR.2019.00704

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics* 38, 4 (2019), 29:1–14. DOI: 10.1145/3306346.3322980

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*. DOI: 10.1007/978-3-030-58452-8_24

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. HoloGAN: Unsupervised learning of 3D representations from natural images. In *ICCV*. DOI: 10.1109/ICCV.2019.00768

Ryan Styles Overbeck, Daniel Erickson, Daniel Evangelakos, Matt Pharr, and Paul Debevec. 2018. A System for Acquiring, Compressing, and Rendering Panoramic Light Field Stills for Virtual Reality. *ACM Transactions on Graphics* 37, 6 (2018), 197:1–15. DOI: 10.1145/3272127.3275031

Albert Parra Pozo, Michael Toksvig, Terry Filiba Schrager, Joyse Hsu, Uday Mathur, Alexander Sorkine-Hornung, Rick Szeliski, and Brian Cabral. 2019. An Integrated 6DoF Video Camera and System Design. *ACM Transactions on Graphics* 38, 6 (2019), 216:1–16. DOI: 10.1145/3355089.3356555

Shmuel Peleg, Moshe Ben-Ezra, and Yael Pritch. 2001. Omnistereo: Panoramic Stereo Imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 3 (2001), 279–290. DOI: 10.1109/34.910880

Federico Perazzi, Alexander Sorkine-Hornung, Henning Zimmer, Peter Kaufmann, Oliver Wang, Scott Watson, and Markus Gross. 2015. Panoramic Video from Unstructured Camera Arrays. *Computer Graphics Forum* 34, 2 (2015), 57–68. DOI: 10.1111/cgf.12541

Christian Richardt. 2020. Omnidirectional Stereo. In *Computer Vision: A Reference Guide*. Springer, 1–4. DOI: 10.1007/978-3-030-03243-2_808-1

Christian Richardt, Peter Hedman, Ryan S. Overbeck, Brian Cabral, Robert Konrad, and Steve Sullivan. 2019. Capture4VR: From VR Photography to VR Video. In *SIGGRAPH Courses*. 1–319. DOI: 10.1145/3305366.3328028

Christian Richardt, Yael Pritch, Henning Zimmer, and Alexander Sorkine-Hornung. 2013. Megastereo: Constructing High-Resolution Stereo Panoramas. In *CVPR*. 1256–1263. DOI: 10.1109/CVPR.2013.166

Christian Richardt, James Tompkin, and Gordon Wetzstein. 2020. Capture, Reconstruction, and Representation of the Visual Real World for Virtual Reality. In *Real VR – Immersive Digital Reality: How to Import the Real World into Head-Mounted Immersive Displays*. Springer, 3–32. DOI: 10.1007/978-3-030-41816-8_1

Ehsan Sayyad, Pradeep Sen, and Tobias Höllerer. 2017. PanoTrace: Interactive 3D Modeling of Surround-View Panoramic Images in Virtual Reality. In *VRST*. DOI: 10.1145/3139131.3139158

Johannes L. Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *CVPR*. 4104–4113. DOI: 10.1109/CVPR.2016.445

Christopher Schroers, Jean-Charles Bazin, and Alexander Sorkine-Hornung. 2018. An Omnistereoscopic Video Pipeline for Capture and Display of Real-World VR. *ACM Transactions on Graphics* 37, 3 (2018), 37:1–13. DOI: 10.1145/3225150

Ana Serrano, Incheol Kim, Zhili Chen, Stephen DiVerdi, Diego Gutierrez, Aaron Hertzmann, and Belen Masia. 2019. Motion parallax for 360° RGBD video. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (2019), 1817–1827. DOI: 10.1109/TVCG.2019.2898757

Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 2020. 3D Photography using Context-aware Layered Depth Inpainting. In *CVPR*. DOI: 10.1109/CVPR42600.2020.00805

Heung-Yeung Shum and Li-Wei He. 1999. Rendering with concentric mosaics. In *SIGGRAPH*. 299–306. DOI: 10.1145/311535.311573

Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. 2018. How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1633–1642. DOI: 10.1109/TVCG.2018.2793599

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. 2019a. DeepVoxels: Learning Persistent 3D Feature Embeddings. In *CVPR*. 2437–2446. DOI: 10.1109/CVPR.2019.00254

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. 2019b. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In *NeurIPS*.

Mel Slater, Martin Usoh, and Anthony Steed. 1994. Depth of Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments* 3, 2 (1994), 130–144. DOI: 10.1162/pres.1994.3.2.130

Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. 2019. Pushing the Boundaries of View Extrapolation With Multiplane Images. In *CVPR*. 175–184. DOI: 10.1109/CVPR.2019.00026

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. 2019. The Replica Dataset: A Digital Replica of Indoor Spaces. (2019). https://github.com/facebookresearch/Replica-Dataset arXiv:1906.05797.

Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. 2019. OpenVSLAM: a Versatile Visual SLAM Framework. In *International Conference on Multimedia*. DOI: 10.1145/3343031.3350539

Richard Szeliski. 2006. Image alignment and stitching: a tutorial. *Foundations and Trends in Computer Graphics and Vision* 2, 1 (2006), 1–104. DOI: 10.1561/0600000009

Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, Rohit Pandey, Sean Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B Goldman, and Michael Zollhöfer. 2020. State of the Art on Neural Rendering. *Computer Graphics Forum* 39, 2 (2020), 701–727. DOI: 10.1111/cgf.14022

Jayant Thatte, Jean-Baptiste Boin, Haricharan Lakshman, and Bernd Girod. 2016. Depth augmented stereo panorama for cinematic virtual reality with head-motion parallax. In *ICME*. DOI: 10.1109/ICME.2016.7552858

Richard Tucker and Noah Snavely. 2020. Single-View View Synthesis with Multiplane Images. In *CVPR*. DOI: 10.1109/CVPR42600.2020.00063

Julien Valentin, Adarsh Kowdle, Jonathan T. Barron, Neal Wadhwa, Max Dzitsiuk, Michael Schoenberg, Vivek Verma, Ambrus Csaszar, Eric Turner, Ivan Dryanovski, Joao Afonso, Jose Pascoal, Konstantine Tsotsos, Mira Leung, Mirko Schmidt, Onur Guleryuz, Sameh Khamis, Vladimir Tankovitch, Sean Fanello, Shahram Izadi, and Christoph Rhemann. 2018. Depth from Motion for Smartphone AR. *ACM Transactions on Graphics* 37, 6 (2018), 193:1–19. DOI: 10.1145/3272127.3275041

Michael Waechter, Mate Beljan, Simon Fuhrmann, Nils Moehrle, Johannes Kopf, and Michael Goesele. 2017. Virtual Rephotography: Novel View Prediction Error for 3D Reconstruction. *ACM Transactions on Graphics* 36, 1 (2017), 8:1–11. DOI: 10.1145/2999533

Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. 2020. BiFuse: Monocular 360 Depth Estimation via Bi-Projection Fusion. In *CVPR*. 462–471. DOI: 10.1109/CVPR42600.2020.00054

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. DOI: 10.1109/TIP.2003.819861

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. 2020. SynSin: End-to-end View Synthesis from a Single Image. In *CVPR*. DOI: 10.1109/CVPR42600.2020.00749

Jianing Zhang, Tianyi Zhu, Anke Zhang, Xiaoyun Yuan, Zihan Wang, Sebastian Beetschen, Lan Xu, Xing Lin, Qionghai Dai, and Lu Fang. 2020. Multiscale-VR: Multiscale Gigapixel 3D Panoramic Videography for Virtual Reality. In *ICCP*. DOI: 10.1109/ICCP48838.2020.9105244

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*. DOI: 10.1109/CVPR.2018.00068

Ke Colin Zheng, Sing Bing Kang, Michael F. Cohen, and Richard Szeliski. 2007. Layered Depth Panoramas. In *CVPR*. DOI: 10.1109/CVPR.2007.383295

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo Magnification: Learning View Synthesis using Multiplane Images. *ACM Transactions on Graphics* 37, 4 (2018), 65:1–12. DOI: 10.1145/3197517.3201323

Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, Federico Alvarez, and Petros Daras. 2019. Spherical View Synthesis for Self-Supervised 360° Depth Estimation. In *3DV*. 690–699. DOI: 10.1109/3DV.2019.00081